```
`timescale 1ns / 1ps

module alu_top_tb;

    reg clk;
    reg rst;
    reg wr_en;
    reg exec_en;
    reg [7:0] instr_in;
    wire [7:0] result;

    // Instantiate DUT
    alu_top uut (
        .clk(clk),
        .rst(rst),
        .wr_en(wr_en),
        .instr_in(instr_in),
        .exec_en(exec_en),
        .result(result)
    );

    // Clock generation
    always #5 clk = ~clk;  // 100 MHz

    // Instruction encoding helper
    function [7:0] encode_instr;
        input [2:0] opcode;
        input [4:0] imm;
        begin
            encode_instr = {opcode, imm};
        end
    endfunction

    // Apply a single instruction: write to FIFO, then execute
    task apply_instruction;
        input [2:0] opcode;
        input [4:0] imm;
        begin
            @(negedge clk);
            instr_in = encode_instr(opcode, imm);
            wr_en = 1;
            @(negedge clk);
            wr_en = 0;
            repeat(2) @(negedge clk);  // wait until written
            exec_en = 1;
            @(negedge clk);
```

```
        exec_en = 0;
        repeat(2) @(negedge clk);  // wait to observe result
    end
endtask

initial begin
    $display("Starting ALU FIFO Test...");
    $dumpfile("alu_top_tb.vcd");
    $dumpvars(0, alu_top_tb);

    clk = 0;
    rst = 1;
    wr_en = 0;
    exec_en = 0;
    instr_in = 8'h00;

    // Reset the system
    repeat(2) @(negedge clk);
    rst = 0;

    // Test sequence
    // ACC starts at 0

    apply_instruction(3'b000, 5'd10);  // ADDI 10   → ACC = 0 + 10 = 10
    $display("ADDI: Result = %d", result);

    apply_instruction(3'b001, 5'd3);   // SUBI 3    → ACC = 10 - 3 = 7
    $display("SUBI: Result = %d", result);

    apply_instruction(3'b010, 5'd5);   // ANDI 5    → ACC = 7 & 5 = 5
    $display("ANDI: Result = %d", result);

    apply_instruction(3'b011, 5'd2);   // ORI 2     → ACC = 5 | 2 = 7
    $display("ORI : Result = %d", result);

    apply_instruction(3'b100, 5'd1);   // XORI 1    → ACC = 7 ^ 1 = 6
    $display("XORI: Result = %d", result);

    apply_instruction(3'b101, 5'd0);   // NOT       → ACC = ~6 = 8'b11111001 =
249
    $display("NOT : Result = %d", result);

    apply_instruction(3'b110, 5'd0);   // INC       → ACC = 249 + 1 = 250
    $display("INC : Result = %d", result);

    apply_instruction(3'b111, 5'd0);   // DEC       → ACC = 250 - 1 = 249
```

```
        $display("DEC : Result = %d", result);

        $display("ALU FIFO Test completed.");
        $finish;
    end

endmodule
```